



Miembros estáticos

Transcripción

[00:00] Entonces, ¿qué tienen en común estos tres objetos diferentes de cuenta? ¿Qué tienen en común? Tienen en común una sola cosa que es esta variable estática. ¿Qué significa? Hablando ya más técnicamente, esta variable, el static, te dice que en un lugar de la memoria va a ser creada esta variable total y todas las demás instancias que sean creadas a partir de esta clase van a ser o van a apuntar mejor dicho aquí, solamente a esta variable total, porque es estática.

[00:43] Al ser estática, no puede ser accedida por la instancia sino es accedida a nivel de clase, por lo cual si yo quisiera aquí decirle `System.out.print de cuenta 3.total`, bueno, primero que todo, no voy a poder acceder porque es privada, entonces yo le podría aquí volver público quizás para el ejemplo, y aquí tengo el total pero está subrayado de amarillo.

[01:20] ¿Qué me está diciendo? Que para yo acceder al total no necesito acceder a la instancia de aquí. ¿Por qué? Bueno, aquí parte de dos puntos. Yo le estoy preguntando a la cuenta 3 por el total. Le podría preguntar a la cuenta 2 o a la cuenta, vamos a llamarla cuenta 1.

[01:42] Yo le podría preguntar: "Cuenta 1, ¿cuántas cuentas hay?" Cuenta 3, ¿cuántas cuentas hay?" Pero eso sería también como, pasando el ejemplo aún más a la vida real, como decirle de repente a ustedes como alumnos no sé: "Juan, ¿cuántos alumnos hay en la clase?" "Pepe, ¿cuántos alumnos hay en la clase" y demás cosas. Y está mal que yo dependa solamente de un alumno para que me responda cuántos alumnos hay en la clase.

[02:12] Sería mucho mejor yo tener un lugar fijo, un contador, digamos, que sea mi lista de asistencia que me diga cuántos alumnos hay en esa clase. Adivinen qué. Es el mismo concepto que vamos a aplicar aquí. ¿Por qué? Porque lo que sería mi lista de asistencia de alumnos sería en este caso la clase cuenta, para tener el total de cuentas. "Diego, ¿qué significa eso?"

[02:41] Significa que yo puedo llamar directamente al total basado en cuenta. Entonces, ¿se podría llamar un atributo directamente de la clase? No tan rápido, estamos llamándolo directamente porque es una variable estática. Por ejemplo, voy a dar enter, control de aquí. Si yo, solamente para este ejemplo, voy a volver público al número, solo para el ejemplo, ¿será que yo puedo acceder aquí, decirle cuenta.número? No puedo. ¿Por qué?

[03:27] Porque número le pertenece a la instancia, no le pertenece a la cuenta. Entonces, son conceptos que ya tenemos que comenzar a separar. Al no ser estático, número vive únicamente dentro de una instancia, no directamente de la clase. ¿Por qué? Porque a este nivel necesita ser estático.

[03:56] Incluso aquí el número me está pidiendo: "Oye, no puedo hacer una referencia a una variable estática desde un contexto que no es estático". Entonces, dime, ¿quieres que yo cambie el número a estático? Bueno, va a depender mucho de la lógica del negocio pero en este caso no va a ser así.

[04:13] Entonces vamos a borrar esta línea, vamos a volverlo nuevamente privado y ya habíamos quedado en encapsular nuestros atributos porque de la misma forma que si mis atributos agencia, saldo, número eran públicos, yo podía manipularlos tranquilamente en cualquier momento.

[04:40] Por ejemplo yo podría hacer algo así como cuenta.total ahora va a ser igual a 7 por ejemplo y yo voy a crear acá tres cuentas. Si yo ejecuto este código, vamos a ver que él me dice, se va creando una cuenta, dos cuentas, tres cuentas, y aquí ¿qué pasó? Me imprimió siete. ¿De dónde? ¿Por qué? Porque yo he manipulado nuevamente ese campo estático. ¿Qué debería hacer ahí?

[05:18] Debería ser mi método getTotal. Para eso primero que todo lo vuelvo privado, al volverlo privado vemos que ya no está compilando aquí el total, obviamente regresamos aquí y vamos a crear nuestro setter. En este caso va a ser un public int getTotal. Perfecto. ¿Qué voy a retornar aquí en getTotal? ¿Voy a retornar this.total? No. ¿Por qué?

[05:58] Porque this hace referencia a este objeto, esta instancia, y ya hemos visto que yo no tengo total a nivel de instancia. Yo lo tengo total a nivel de clase. Yo podría hacer algo así como cuenta.total y dejarlo así tranquilamente. Entonces, de esta forma yo ya estoy accediendo a mi variable getTotal desde un método.

[06:29] Pero ahora nuevamente entro al punto anterior. ¿Por qué? Este método getTotal sí hace referencia a un método de mi instancia. Yo no puedo hacer aquí un cuenta.getTotal. ¿Por qué? Porque esta variable aquí se refiere a la instancia. Es un método de mi instancia y no es un método estático. ¿Qué debería hacer yo para solucionar este problema? Muy fácil. Vuelvo el método estático.

[07:07] Entonces ahora este método ya es accesible desde mi clase cuenta, ya se volvió un método de la clase y no de la instancia. Eso es algo que tenemos que tener muy claro. Esa es nuestra instancia y esta sería la clase. La clase tiene sus propios métodos, sus propios atributos. Ahora con esto yo ya estoy asegurando que yo no puedo manipular el total directamente, por ejemplo.

[07:40] Si yo hago aquí un cuenta.total vemos que para comenzar Eclipse no encuentra la variable y solamente tengo acceso a total. Solo puedo tener lectura en esta variable total. No puedo igualarla a ningún valor porque no me es permitido.

